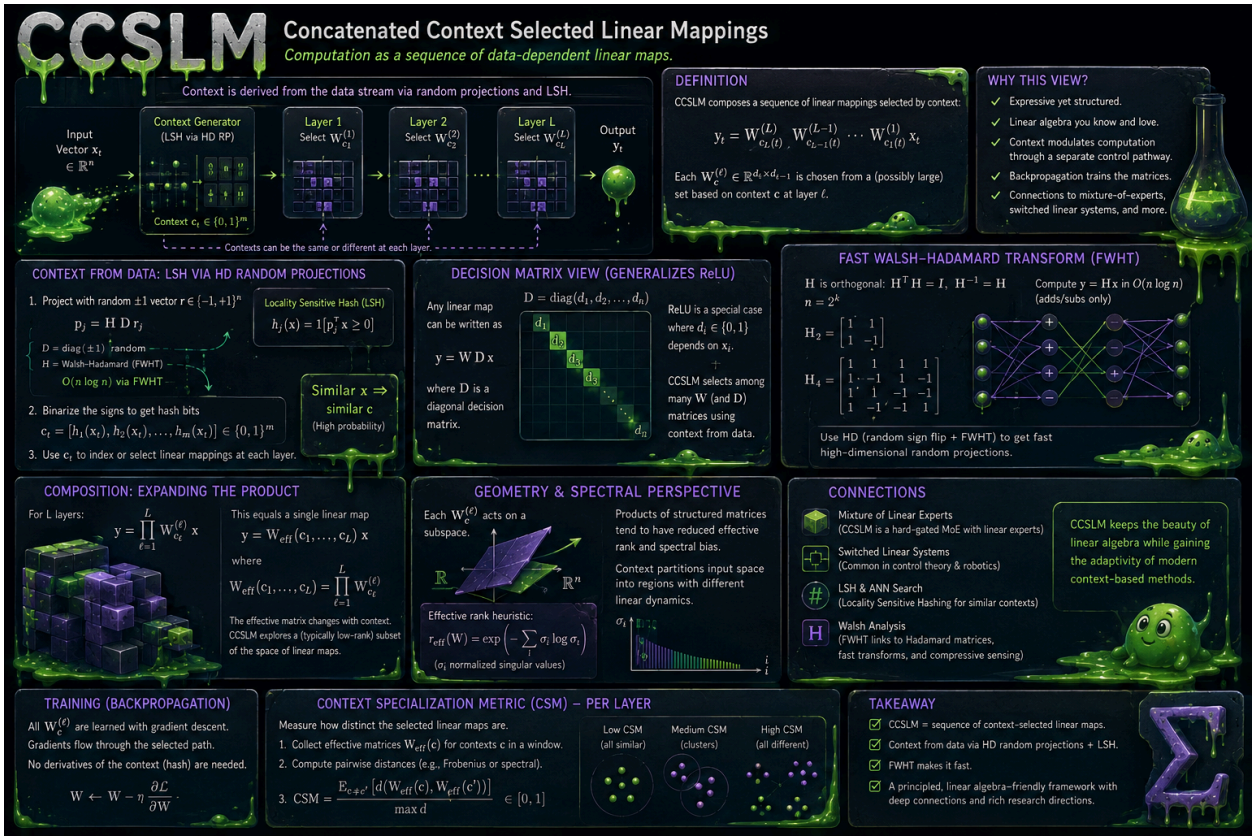
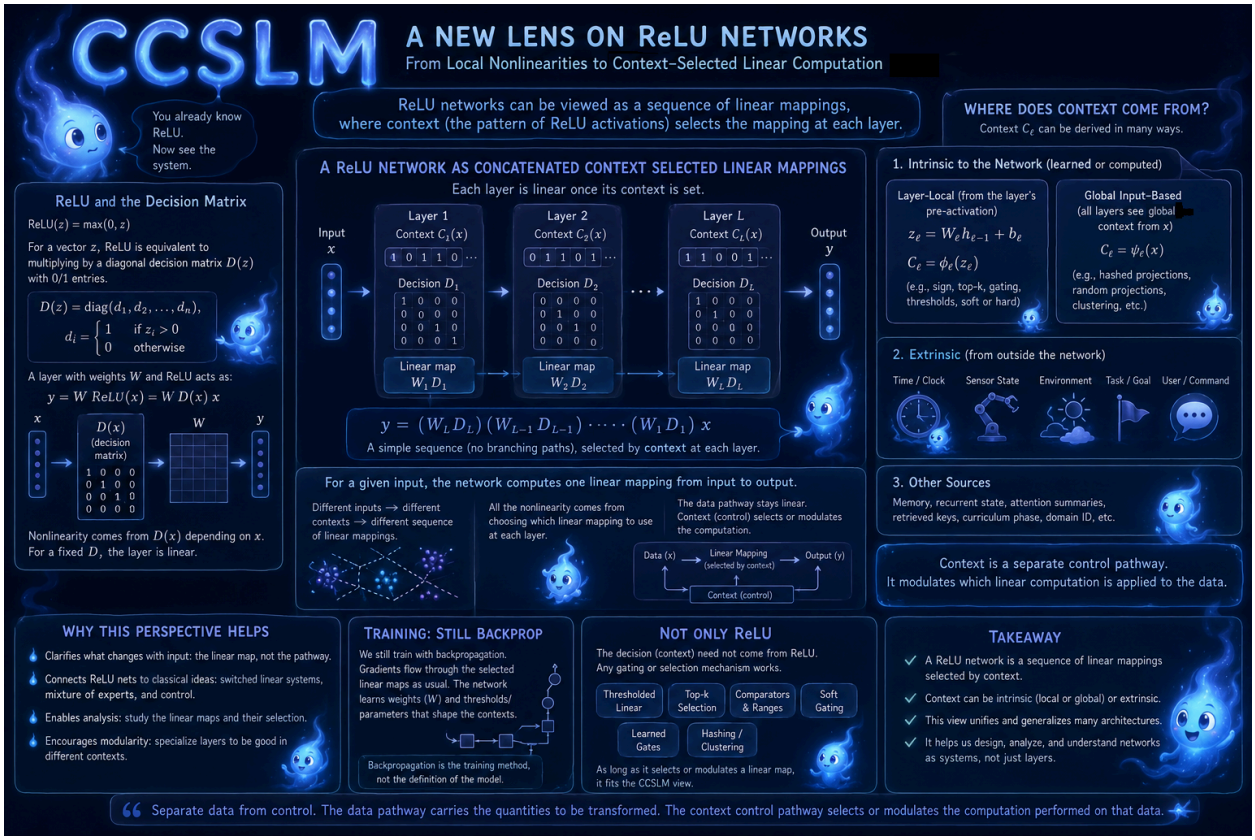
CCSLM
CONCATENATED CONTEXT SELECTED LINEAR MAPPINGS
Context modulates computation through a separate control pathway.
Each layer has its own independent context.
Different layers can respond to different aspects of the world.
CONTEXT / CONTROL PATHWAY
Provides persistent state that selects the computation.
Time of Day
Robot Speed
Battery Level
Goal Mode
Terrain Type
Hysteresis State
Sun Position
PRINCIPLE:
SEPARATE DATA FROM CONTROL.
DATA PATHWAY: quantities to be transformed
CONTROL PATHWAY: persistent state that selects or modulates the computation
DATA PATHWAY
LAYER 1 (Select W_1)
LAYER 2 (Select W_2)
LAYER 3 (Select W_3)
LAYER N (Select W_N)
Input Vector x
Output Vector y
$y = W_N^{k_N}(c_N)\, W_{N-1}^{k_{N-1}}(c_{N-1}) \cdots W_2^{k_2}(c_2)\, W_1^{k_1}(c_1)\, x$
A simple sequence of chosen linear mappings chained in series. No multiple different pathways.
THINK SLIME
Slime adapts, flows, and changes form based on context... but stays slime.
WHAT IS CCSLM?
A system that performs a computation by concattnating (chaining) linear mappings, where the specific linear mapping used at each stage is selected by context.
EXAMPLE CONTEXT EFFECTS
06:00 → 12:00 → 18:00 → 24:00
Different Linear Mapping
Different Linear Mapping
Different Linear Mapping
Different Linear Mapping
TRAINING (optional)
Mappings can be learned using backpropagation (or other methods). Backpropagation is used only as s training method, not part of the runtime computation.
KEY INSIGHT
Context selects WHICH linear mapping to use.
Linear mappings transform WHAT the data is.
Control and data stay separate.
WHY CCSLM?
✓ Modular
✓ Interpretable
✓ Efficient
✓ Robust to noise
✓ Natural fit for real world stateful systems

Robotic version


CCSLM
Concatenated Context Selected Linear Mappings
Computation as a sequence of data-dependent linear maps.
Context is derived from the data stream via random projections and LSH.
Input Vector $x_t \in \mathbb{R}^n$
Context Generator (LSH via HD RP)
Context $c_t \in \{0,1\}^m$
Layer 1 Select $W_{c_1}^{(1)}$
Layer 2 Select $W_{c_2}^{(2)}$
Layer L Select $W_{c_L}^{(L)}$
Output y_t
Contexts can be the same or different at each layer.
DEFINITION
CCSLM composes a sequence of linear mappings selected by context:
$y_t = W_{c_L(t)}^{(L)} W_{c_{L-1}(t)}^{(L-1)} \cdots W_{c_1(t)}^{(1)} x_t$
Each $W_c^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ is chosen from a (possibly large) set based on context c at layer ℓ.
WHY THIS VIEW?
Expressive yet structured.
Linear algebra you know and love.
Context modulates computation through a separate control pathway.
Backpropagation trains the matrices.
Connections to mixture-of-experts, switched linear systems, and more.
CONTEXT FROM DATA: LSH VIA HD RANDOM PROJECTIONS
1. Project with random ± 1 vector $r \in \{-1,+1\}^n$
$p_j = H D r_j$
D = diag(±1) random
H = Walsh-Hadamard (FWHT)
$O(n \log n)$ via FWHT
Locality Sensitive Hash (LSH)
$h_j(x) = 1[p_j^T x \geq 0]$
2. Binarize the signs to get hash bits
$c_t = [h_1(x_t), h_2(x_t), \ldots, h_m(x_t)] \in \{0,1\}^m$
3. Use c_t to index or select linear mappings at each layer.
Similar x ⇒ similar c
(High probability)
DECISION MATRIX VIEW (GENERALIZES ReLU)
Any linear map can be written as
$y = W D x$
where D is a diagonal decision matrix.
$D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$
ReLU is a special case where $d_i \in \{0,1\}$ depends on x_i.
CCSLM selects among many W (and D) matrices using context from data.
FAST WALSH–HADAMARD TRANSFORM (FWHT)
H is orthogonal: $H^T H = I$, $H^{-1} = H$
$n = 2^k$
Compute $y = Hx$ in $O(n \log n)$ (adds/subs only)
$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$
Use HD (random sign flip + FWHT) to get fast high-dimensional random projections.
COMPOSITION: EXPANDING THE PRODUCT
For L layers:
$y = \prod_{\ell=1}^{L} W_{c_\ell}^{(\ell)} x$
This equals a single linear map
$y = W_{eff}(c_1, \ldots, c_L)\, x$
where
$W_{eff}(c_1, \ldots, c_L) = \prod_{\ell=1}^{L} W_{c_\ell}^{(\ell)}$
The effective matrix changes with context. CCSLM explores a (typically low-rank) subset of the space of linear maps.
GEOMETRY & SPECTRAL PERSPECTIVE
Each $W_c^{(\ell)}$ acts on a subspace.
R
$\mathbb{R}^n$
Products of structured matrices tend to have reduced effective rank and spectral bias.
Context partitions input space into regions with different linear dynamics.
Effective rank heuristic:
$r_{eff}(W) = \exp\left(-\sum_i \sigma_i \log \sigma_i\right)$
(σ_i normalized singular values)
σ_i
i
CONNECTIONS
Mixture of Linear Experts
(CCSLM is a hard-gated MoE with linear experts)
Switched Linear Systems
(Common in control theory & robotics)
LSH & ANN Search
(Locality Sensitive Hashing for similar contexts)
H
Walsh Analysis
(FWHT links to Hadamard matrices, fast transforms, and compressive sensing)
CCSLM keeps the beauty of linear algebra while gaining the adaptivity of modern context-based methods.
TRAINING (BACKPROPAGATION)
All $W_c^{(\ell)}$ are learned with gradient descent.
Gradients flow through the selected path.
No derivatives of the context (hash) are needed.
$W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W}$
CONTEXT SPECIALIZATION METRIC (CSM) – PER LAYER
Measure how distinct the selected linear maps are.
1. Collect effective matrices $W_{eff}(c)$ for contexts c in a window.
2. Compute pairwise distances (e.g., Frobenius or spectral).
3. $\mathrm{CSM} = \frac{\mathbb{E}_{c \neq c'}[d(W_{eff}(c), W_{eff}(c'))]}{\max d} \in [0,1]$
Low CSM (all similar)
Medium CSM (clusters)
High CSM (all different)
TAKEAWAY
CCSLM = sequence of context-selected linear maps.
Context from data via HD random projections + LSH.
FWHT makes it fast.
A principled, linear algebra–friendly framework with deep connections and rich research directions.
Σ

Linear Algebra version.


CCSLM
A NEW LENS ON ReLU NETWORKS
From Local Nonlinearities to Context-Selected Linear Computation
You already know ReLU. Now see the system.
ReLU networks can be viewed as a sequence of linear mappings, where context (the pattern of ReLU activations) selects the mapping at each layer.
WHERE DOES CONTEXT COME FROM?
Context C_ℓ can be derived in many ways.
ReLU and the Decision Matrix
$\mathrm{ReLU}(z) = \max(0, z)$
For a vector z, ReLU is equivalent to multiplying by a diagonal decision matrix $D(z)$ with 0/1 entries.
$D(z) = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, $d_i = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{otherwise} \end{cases}$
A layer with weights W and ReLU acts as:
$y = W\,\mathrm{ReLU}(x) = W\,D(x)\,x$
x
D(x) (decision matrix)
W
y
Nonlinearity comes from $D(x)$ depending on x. For a fixed D_i, the layer is linear.
A ReLU NETWORK AS CONCATENATED CONTEXT SELECTED LINEAR MAPPINGS
Each layer is linear once its context is set.
Input x
Layer 1
Context $C_1(x)$
1 0 1 1 0 ⋯
Decision D_1
Linear map $W_1 D_1$
Layer 2
Context $C_2(x)$
0 1 1 0 1 ⋯
Decision D_2
Linear map $W_2 D_2$
Layer L
Context $C_L(x)$
1 1 0 0 1 ⋯
Decision D_L
Linear map $W_L D_L$
Output y
$y = (W_L D_L)(W_{L-1} D_{L-1}) \cdots (W_1 D_1)\,x$
A simple sequence (no branching paths), selected by context at each layer.
For a given input, the network computes one linear mapping from input to output.
Different inputs → different contexts → different sequence of linear mappings.
All the nonlinearity comes from choosing which linear mapping to use at each layer.
The data pathway stays linear. Context (control) selects or modulates the computation.
Data (x)
Linear Mapping (selected by context)
Output (y)
Context (control)
1. Intrinsic to the Network (learned or computed)
Layer-Local (from the layer's pre-activation)
$z_\ell = W_\ell h_{\ell-1} + b_\ell$
$C_\ell = \phi_\ell(z_\ell)$
(e.g., sign, top-k, gating, thresholds, soft or hard)
Global Input-Based (all layers see global context from x)
$C_\ell = \psi_\ell(x)$
(e.g., hashed projections, random projections, clustering, etc.)
2. Extrinsic (from outside the network)
Time / Clock
Sensor State
Environment
Task / Goal
User / Command
3. Other Sources
Memory, recurrent state, attention summaries, retrieved keys, curriculum phase, domain ID, etc.
Context is a separate control pathway.
It modulates which linear computation is applied to the data.
WHY THIS PERSPECTIVE HELPS
Clarifies what changes with input: the linear map, not the pathway.
Connects ReLU nets to classical ideas: switched linear systems, mixture of experts, and control.
Enables analysis: study the linear maps and their selection.
Encourages modularity: specialize layers to be good in different contexts.
TRAINING: STILL BACKPROP
We still train with backpropagation. Gradients flow through the selected linear maps as usual. The network learns weights (W) and thresholds/parameters that shape the contexts.
Backpropagation is the training method, not the definition of the model.
NOT ONLY ReLU
The decision (context) need not come from ReLU. Any gating or selection mechanism works.
Thresholded Linear
Top-k Selection
Comparators & Ranges
Soft Gating
Learned Gates
Hashing / Clustering
As long as it selects or modulates a linear map, it fits the CCSLM view.
TAKEAWAY
A ReLU network is a sequence of linear mappings selected by context.
Context can be intrinsic (local or global) or extrinsic.
This view unifies and generalizes many architectures.
It helps us design, analyze, and understand networks as systems, not just layers.
Separate data from control. The data pathway carries the quantities to be transformed. The context control pathway selects or modulates the computation performed on that data.

Deep Learning version.